NASA CONTRACTOR
REPORT

NASA CR-144031

(NASA-CR-144031) BANNING DESIGN AUTOMATION SOFTWARE IMPLEMENTATION Final Report (M&S Computing, Inc.) 41 p HC $4.00 CSCL 09B

N76-22943

Unclas

G3/61 25279

BANNING DESIGN AUTOMATION SOFTWARE IMPLEMENTATION

By R. L. Kuehlthau

M&S Computing, Inc.
Huntsville, Alabama 35805

October 28, 1975

Prepared for

NASA - GEORGE C. MARSHALL SPACE FLIGHT CENTER
Marshall Space Flight Center, Alabama 35812

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO. CR-144031	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
4. TITLE AND SUBTITLE Banning Design Automation Software Implementation		5. REPORT DATE October 28, 1975
		6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) R. L. Kuehlthau		8. PERFORMING ORGANIZATION REPORT # 75-0037
9. PERFORMING ORGANIZATION NAME AND ADDRESS M&S Computing, Inc. P.O. Box 5183 Huntsville, Alabama		10. WORK UNIT NO.
		11. CONTRACT OR GRANT NO. NAS8-25621
		13. TYPE OF REPORT & PERIOD COVERED CONTRACTOR REPORT
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration Washington, D. C. 20546		14. SPONSORING AGENCY CODE
15. SUPPLEMENTARY NOTES		

16. ABSTRACT

This report formally documents the efforts performed by M&S Computing, Inc., under Contract No. NAS8-25621, "Banning Design Software, Implementation," for the George C. Marshall Space Flight Center of the National Aeronautics and Space Administration. The work was administered under the technical direction of the Astrionics Laboratory of the George C. Marshall Space Flight Center with Mr. John Gould acting as project engineer.

17. KEY WORDS		18. DISTRIBUTION STATEMENT Unclassified - Unlimited	
19. SECURITY CLASSIF. (of this report) Unclassified	20. SECURITY CLASSIF. (of this page) Unclassified	21. NO. OF PAGES 41	22. PRICE NTIS

1. INTRODUCTION

This final report summarizes the work done by M&S Computing under Contract No. NAS8-25621. It also provides documentation of numerous efforts which have not been directly documented with programmer or user manuals. The overall objective of this contract has been to design, implement, and support a system of computer programs which would aid engineers in the design, fabrication, and testing of large scale integrated circuits (LSI's), hybrid circuits, and printed circuit boards (PCB's). Table 1-1 lists the programs which have been implemented according to function. These programs comprise the final design automation software system. Many of them are improved versions of software written or implemented in the earlier stages of the contract. All programs operate on the Sigma 5 or Sigma 2 computer systems, and are written almost entirely in FORTRAN. Many have been widely distributed to other Government and industry users.

The development of each functional group of programs into the final release is discussed in Section 5 of this report. The last section summarizes the general support efforts which have been performed under this contract.

Most major programs are documented by formal reports which are referenced and explicitly identified in the references attached to this report. Other documentation which was previously released informally has been formalized by inclusion in the appendices of this report.

IMPLEMENTED PROGRAMS

Automatic Layout Programs

PRF - PMOS and CMOS LSI Circuit Layout
PR2D - CMOS and SOS LSI Circuit Layout
PCB - Hybrid and Printed Circuit Layout

Analysis Programs

LOGSIM - Logic Simulation Program
FETSIM - FET Circuit Simulation (Analog)
FETLOG - Combined Logic and Circuit Simulation
LASAR - Test Pattern Generation Program
MAP - Mask Analysis Program

Graphics Display Program

AIDS - Interactive Graphics Display and Editing
DSPLIB - Display Librarian

Interface Programs

ARTWORK - Gerber Plotter Mask Generation
MANART - Mann Pattern Generator Artwork Generation
PCBART - PCB Mask Interface to Gerber Plotter
GCCP - Interface to Gerber 700 Plotter
PRFAID - Interface into AIDS Graphics Display System
CARD - Interface Between LASAR and H316 Tester
MACARD - Interface to Macrodata Tester
NTRAN - Network Translator Between PRF, LASAR, FETLOG, and MAP
EDIT - Card Image Edit Utility Program

Table 1-1

2. AUTOMATIC LAYOUT PROGRAMS

Use of the computer to place electronic elements and route electrical connections between them has tremendously reduced the effort required to layout artwork masks for LSI circuits and PCB's. The layout problem consists of determining a good placement of the elements to be interconnected and routing signal paths which implement the circuit specified by the designer. The digital placement and routing data can then be processed by the interface programs into a format suitable for driving an automatic photoplotter to produce the required artwork. The LSI and PCB environments are substantially different in the types of elements to be interconnected, the amount of freedom permitted in placement and routing, and in the objectives of the placement and routing algorithms.

2.1 PRF

The Banning PRF Program, originally written for a GE635 computer, has undergone major revisions under this contract but remains functionally unchanged. M&S Computing's improvements include:

- elimination of numerous program bugs which caused program termination or errors during execution,
- overlaying and redesign of coding to make maximum use of the available memory,
- implementation of CMOS technology overlays,
- adaptation for MSFC border, mask, and chip identification, and
- implementation of packed data arrays to increase program capacity.

Complete documentation of the program has been provided by the PRF Programmer's and User's Manuals (see References 1 and 2). Revisions to the program have been documented in updates to the User's Manual. This program is currently the most reliable of the layout programs and is the only program supporting PMOS technology. Manual editing of its output is still required, however, to utilize chip area more efficiently or correct an occasional error.

2.2 PR2D

The PR2D Program is a newer version of the PRF Layout Program employing more sophisticated algorithms, particularly in routing. This

program was written by RCA for a Spectra 70 computer and has been undergoing frequent updates for the past several years. To facilitate implementation of the updates, the EDIT utility processor was written to translate source tapes into Sigma 5 code. The current version of PR2D supports metal gate, silicon gate, and silicon-on-sapphire technologies. It is documented primarily by a thoroughly commented listing which assures accuracy but requires a relatively sophisticated user. It is recommended that a user's manual be prepared for the PR2D Program once it has become sufficiently stable. Like PRF, PR2D layouts will normally require some manual editing on the graphics display system.

2.3 PWB

The PWB and hybrid circuit layout capability is actually implemented in a package of four programs which are executed in sequence. The first of these is the input processor which accepts designer inputs in a convenient free format. This processor, which was written entirely under this contract, maintains a library of circuit board components, their geometries, and their electrical characteristics. The three remaining PWB Programs were converted from programs originally written by General Electric. They have been converted into compatible code, sized for execution in available memory, and extensively tested using both printed circuit and hybrid circuit test cases. The routing program has been modified to include the capability to route various width lines, provide either a feedthrough or dielectric isolation mask, and to provide output which is compatible with the graphic layout editor and the plotter interface software. The GCCP Program (Appendix A) has been written to provide an interface between these programs and the Gerber 700 plotter used by MSFC to plot PWB artwork masters. The PWB Program package has been documented by the PWB User's and Programmer's Manuals (see References 3 and 4). Revision 1 to the User's Manual reflects the current status of this package.

The results using the PWB system have been highly dependent on the complexity and density of the circuit. The best results in two-layer PWB routing were obtained with manually routed power and ground and with vertical and horizontal segments segregated to different levels. The program's most serious flaw is its tendency to block pins in the early stages of wiring. This is particularly serious in hybrid circuits mounting LSI chips. It is recommended that future routing algorithms be implemented in a semi-interactive environment in which partially complete routing may be examined, modified with an interactive graphics system, and restarted with a minimum of user effort.

3. ANALYSIS PROGRAMS

Numerous analysis programs have been studied and implemented in the course of this contract. Most of the programs in current use were either selected from among other currently available programs or designed after analyzing the strong and weak points of their predecessors. The NTRAN Interface Program, to be discussed in Section 4, links these programs with the chip layout programs to provide a powerful analysis system.

3.1 LOGSIM

The LOGSIM Logic Simulator was derived from studies of the RCA LOGSIM III and IV Simulators. It is a large capacity event simulator with the capability to accurately simulate the effects of unknown states, rise and fall times, and floating nodes in MOS logic circuits. In addition to models for common logic elements, LOGSIM permits the user to define ROM circuits and individually define element propagation delays in his model. Bit packing is employed in simulator arrays to achieve the high program capacity. The LOGSIM User's and Programmer's Manuals (see References 5 and 6) provide full documentation of these programs.

3.2 FETSIM

Analog circuit simulation is provided by the FETSIM Program. This program, which was written initially by RCA, uses a sophisticated MOS transistor model in a network analysis program which adjusts the time interval of integration dynamically to maintain accuracy. The coding of this program has been improved by removing non-essential calculations from within loops, reducing execution time significantly. FETSIM is currently compatible with PMOS, NMOS, bulk CMOS, and SOS CMOS technologies.

3.3 FETLOG

The FETLOG Performance Simulator integrates the speeded FETSIM Program into LOGSIM so that accurate analog simulations of circuit elements can be driven directly by logic elements of the circuit. Logic gates connected to the analog circuits are driven by logic events interpreted from the analog node voltages. By breaking large circuits into a number of smaller disjunct circuits interfaced by gate signals, FETLOG may be used to provide reasonably accurate analog simulations in a fraction of the time normally required. FETLOG has the capability to determine the logical operation of disjunct circuits using analog simulations and automatically enter the result in truth tables so that subsequent simulation will be logical. The FETLOG User's Manual (see Reference 7) documents operation of this program.

3.4 LASAR

The LTV originated LASAR package is a system of four programs which provide automatic generation and verification of logic test patterns to detect faults in logic circuitry. Other programs which were investigated to perform this function include RCA's AGAT, FLTSIM, and WORSTSIM and Telpar's CTGP. The most significant features of the LASAR package are its capability to correctly generate tests for sequential circuits and its relatively large capacity.

The LASAR Stimulus Generation Program is quite complex and was studied in detail in an attempt to improve the percentage of faults detected and reduce execution time. Several program bugs were uncovered and a number of improvements implemented including the capability to define initial node states, illegal node configurations, and race detection criteria. A major modification, termed Gathered Loop Processing (GLOP), was implemented to streamline the process by which the stimulus generator detects the existence of latched loops. This change results in significantly reduced execution time for logic having many loops. Appendix B documents these program modifications.

The last program of the LASAR package is a fault simulator which provides an accurate indication of the percentage of single failure faults detected by the test pattern set. Interface programs have been developed to drive both the H316 and Macrodata testers with LASAR test patterns.

The experience gained in using the TPG Programs on a number of TTL logic boards and LSI circuits indicates that the user must apply it carefully. Excessive run times will be incurred if large global feedback loops or uninitialized ripple counters are present. The programs are inherently limited to static synchronous logic (no one-shots or edge triggered flip-flops) and do not have an efficient technique to force toggling of the clock signals.

3.5 MAP

The Mask Analysis Program (MAP) has been developed as a tool to determine the electrical and logical characteristics of a given set of mask artwork and design rules. It supersedes the Macrodata Artcheck, Operational Equation Generation, and Crosstalk Capacitance Calculation Programs implemented in the early phases of this contract. The primary disadvantages of the Macrodata Programs are their lack of flexibility in the specification of design rule checks and their limited capacity. MAP has been designed to perform all of the functions of the Macrodata Programs according to a user-generated

file of instructions. The instruction file will, in general, be technology-dependent and may be changed to trade-off processing time with the thoroughness of design rule checks. The speed and capacity of the MAP Program are achieved by simplifying all geometry to rectangles and keeping the bulk of the mask data on disk.

The MAP Program is interfaced to the graphics data base in the PRF format. Limited on-line design rule checks of artwork displayed on the Sigma 2 AIDS is accomplished by executing a Sigma 2 version of MAP as an AIDS overlay. MAP is also interfaced into FETLOG and LASAR through the NTRAN Program to permit simulation and test pattern generation of the logic it traces. The MAP User's Manual documents program operation.

4. INTERACTIVE GRAPHICS DISPLAY SYSTEMS

Since most layouts generated by the Automatic Layout Programs could be improved with manual editing, an effective means of performing and displaying the results of edits was essential. Initial display system development performed on the Sanders refresh display was moved to the Computek storage display terminal on the Sigma 2 when it became evident that refresh displays would flicker objectionably when complex LSI masks were viewed. The Artwork Interactive Design System (AIDS) currently operating on the Sigma 2 was developed in several iterations between software design and application. The 24K word memory limitation and the desire to implement the program in FORTRAN made this task significantly more difficult than originally envisioned. Speed of storage display repaint became the objective of numerous program rewrites and improvements, with some of the FORTRAN routines being rewritten in assembly language. The final program has been highly optimized for both minimum memory and minimum repaint time. Since only six memory words remain in one of the program overlays, future program modifications will be difficult unless additional memory can be provided. Program documentation is provided by the AIDS Program Description (see Reference 9).

A Sigma 5 time-sharing version of the AIDS was implemented to facilitate editing of PCB and hybrid circuits. Although this version was fully operational, the repaint time of its display was restricted by operating system constraints to impractically long periods. PCB and hybrid circuit editing was, therefore, moved to the Sigma 2 AIDS.

5. INTERFACE PROGRAMS

Much of the work under this contract has been to integrate programs obtained from various industry and Government sources into a complementary system of programs. This usually involved either modifying the original programs or generating interface programs to translate between data bases.

5.1 ARTWORK

ARTWORK translates PRF format line, shape, and cell data into Gerber 1200 plotter commands. The original program has undergone only minor changes which support cells containing the flash apertures commonly used in PC board artwork. The current version accepts flash apertures in the cell library definitions and produces the appropriate flashes in the Gerber 1200 output. ARTWORK User's and Programmer's Manuals (see References 10 and 11) serve as program documentation.

5.2 MANART

The MANART Program converts Gerber 1200 data into commands to produce artwork on the D. W. Mann Pattern Generator. Output is sorted on both X and Y. The MANART User's Manual provides program documentation.

5.3 PCBART

This program is part of the PWB package and was w· tten to convert the PWB router output into commands for the Gerber 700 plotter. It supports various width lines, user-specified pad location and orientation, and automatic generation of board outline and alignment masks.

5.4 GCCP

The GCCP Program converts Gerber 1200 data into Gerber 700 commands. Appendix A documents this program.

5.5 PRFAID

All layout program output and the graphic output of MAP are in PRF card image format for graphic data. These outputs may be displayed and edited on the AIDS by translating them with the PRFAID Program. Options are also available to provide reverse translation into PRF format once the file has been edited. The PRFAID Programs are documented in the AIDPRF/ PRFAID User's Manual.

5.6 CARD

The CARD Program is a part of the LASAR package which is used to translate test patterns into a card format suitable for driving the H316 circuit board tester.

5.7 MACARD

MACARD translates test patterns into a magnetic tape which will drive the Macrodata tester. Although the program has been coded, the Macrodata tester has not been available to check for proper operation of the generated tape.

5.8 NTRAN

NTRAN was developed to provide the users of the MSFC Banning Design Automation System a means for generating input for the FETLOG and LASAR Programs that is representative of the logic circuitry included in a layout produced by the PRF or PR2D Programs. NTRAN utilizes a PRF or PR2D input deck and a "library" file to produce output formatted as input for either of the target programs (FETLOG or LASAR). The library file consists of a model in terms of the target program for each of the standard patterns utilized in the design. The library file is in card image format and is easily prepared and maintained utilizing standard system software.

NTRAN was designed to maintain, to the extent possible, the original identity of the information as it is transformed into the required form. This goal was realized by adopting a "cell-pin" convention for the signal names in the network. The "cell" portion of the name indicates the PRF (or PR2D) cell number that the signal is associated with and the "pin" identifies the pin name within the pattern type that the signal is associated with. The definition of a pin is extended to include all logic nodes of the pattern type whether the node is a true pin or not. A convention is employed that allows the user to readily ascertain the nature of the pin by using numeric names for true pins and non-numeric names for nodes that are internal to the pattern. For example, signal 100-AA is recognized as an internal node of PRF cell number 100. The "cell-pin" signal name convention is maintained throughout all phases of the process for FETLOG translations, but must be abandoned for the final output of LASAR translations.

Since neither FETLOG nor LASAR is programmed to handle the unique properties of transmission gate logic, NTRAN generates a reasonble approximation of a transmission gate network with a ROM for FETLOG and a sum of products network for LASAR. A transmission gate network consisting of as many as 15 transmission gates can be handled by the program.

A second form of NTRAN translation interfaces the output of the Mask Analysis Program to the FETLOG and LASAR Programs. A logical description of the masks is available in the operational equations. Equivalent NAND or ROM models can be generated for LASAR or FETLOG analysis. In addition, the user may specify disjunct circuits for which NTRAN will generate FETLOG disjunct circuit data from the operational equation, capacitance calculation, and transistor characterization data. The NTRAN Program is documented by the NTRAN User's Manual (see Reference 12).

5.9 EDIT

The EDIT Program performs line editing of card images to or from blocked or unblocked tapes. It has been extensively used as a utility in the maintenance of source files, the deblocking and blocking of tapes, and as a vehicle for translation of source files into compatible code. The EDIT User's Manual in Appendix C documents this program.

6. SYSTEM SUPPORT

Support activities under this contract included assisting NASA and other users of the system with control and data deck setup, correction of program errors as they were identified, and maintenance of hardware and software compatibility. Current programs have, on the average, required source updates once a year and load module generation roughly four times a year. Several major routing projects were also undertaken by M&S Computing personnel to demonstrate the operation of the PWB Programs. These projects required the generation of module libraries as well as the coding of normal data cards.

APPENDIX A

GCCP USER'S GUIDE

APPENDIX A

GERBER COMMAND CONVERSION PROGRAM - USER'S GUIDE

A.1 DATA CARD SETUP (see Figure A-1)

A.1.1 Option Card

1. **Form:** OPTN bb a,b,c,d,e

   Columns 1-4 - OPTN

   Beg. Col. 7 - five option values separated by commas with no internal blanks (a, b, c, d, e).

2. Content:

   Option a - print option: a = 0 = off

   a = 1 = all input and output commands printed.

   Option b - punch option: b = 0 = off

   b = 1 = punch output commands.

   Option c - c = number of levels to skip on input data before beginning to process.

   Option d - d = number of levels to process, or

   d = 0 = all levels will be processed until a double end-of-file is encountered.

   Option e - e = number of leading digits to ignore on x-y data, where:

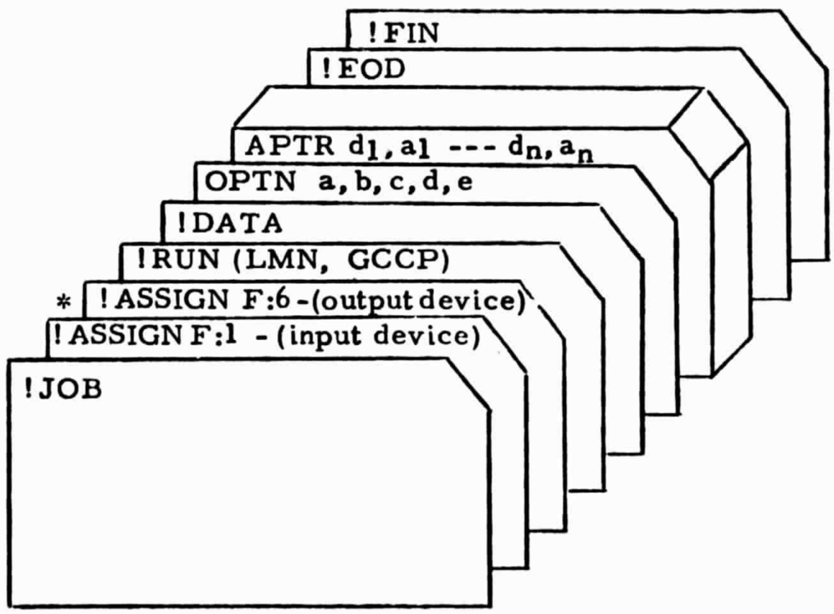
   e = 0 = no digits dropped,

   e = 1 = first digit dropped, scales data by 10,

   e = 2 = first two digits dropped, scales data by 100.

3. Default: no option card = OPTNbb1, 0,0,0,0

GCCP RUNDECK



*** Suggested Assignment**

!ASSIGN F:6, (Device, CPA04), (FBCD)

Figure A-1

APERTURE SELECTION FOR WHEEL 2

APT 1	.025 R	LINE
APT 2	.036 R	
APT 3	.062 R	
APT 4	.075 R	
APT 5	.010 S	
APT 6	.015 S	
APT 7	.020 X .005	
APT 8	.005 X .020	
APT 9	.033 X .075	FLASH
APT 10	.075 X .033	
APT 11	.110 X .035	
APT 12	.035 X .110	
APT 13	.050	
APT 14	.062	
APT 15	.070	
APT 16	.075 X .016	
APT 17	.090 X .020	
APT 18	.125 X .020	
APT 19	.060 SQ	
APT 20	.025 SQ	LINE
APT 21	.025 X .010	FLASH
APT 22	MOIRE	
APT 23	.085 X .020	HORSESHOE
APT 24	.187 OD MOIRE	

A.1.2 Aperture Replacement Cards

1. Form: APTRbbd$_1$,a$_1$,d$_2$,a$_2$,... ...d$_n$,a$_n$

   Columns 1-4 - APTR

   Beg. Col. 7 - up to ten value pairs (d_n, a_n), where all values are separated by commas with no internal blanks.

   Any number of APTR cards are allowed.

2. Content:

   Aperture replacement pairs - d_n, a_n = when aperture d_n is encountered in input commands, it is replaced with aperture a_n in output commands.

3. Default: no aperture replacement $a_n = d_n$.

APPENDIX B

LASAR MODIFICATIONS

LASAR PROGRAM DATA FILE INTERFACE

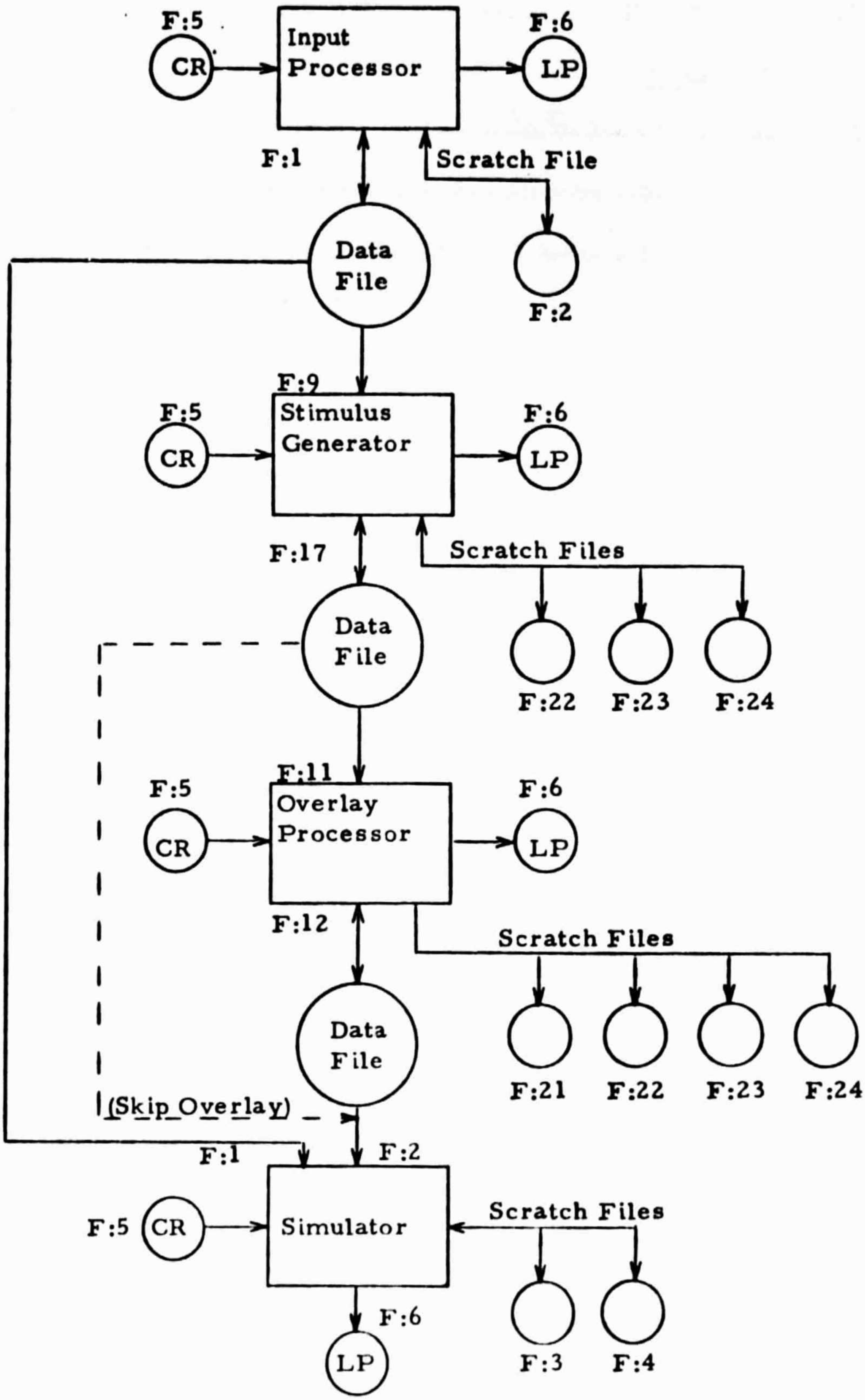


Figure B-1

B.1 STIMULUS GENERATION DATA CARDS

B.1.1 Run Option Card

Col. 1-4 - Primary print option:

=0 - stimuls patterns are printed.

=1 - same as 0 plus new components detected for each set printout.

=2 - same as 1 plus loop data printout.

=3 - same as 2 plus configuration table printout.

Col. 5-8 - Secondary print option:

=0 - off.

=1 - component nodes not detected are printed at end of run.

=2 - same as 1 plus miscellaneous configuration data printout.

Col. 9-12 - Maximum number of stimulus patterns per set:

=1-600 - maximum number of stimulus patterns which may be generated for any set.

≤ 0 - reset to 600.

<600 - reset to 600.

Col. 13-16 - Maximum number of nodes per loop:

>0 - maximum number of nodes per loop.

≤ 0 - reset to 2.

When a loop is found with more than the maximum number of nodes, the loop is not initialized and a warning message is printed.

Col. 17-20 - User's node number of first node for which stimuli are to be generated.

=0 - processing will begin on the first output node and all output nodes will be processed.

>0 - processing will begin on that user's node and all nodes following (i.e., with higher program node numbers) will also be processed. If a non-existent user's node number is given, an error termination occurs.

<0 - output response patterns will be taken from INIT card inputs.

Col. 21-24 - Loop reversing option:

=2 - on, an attempt is made to unlatch any latched loops by generating initials for the next level in the reverse states.

≠2 - off.

Col. 25-28 - Allow criticality gaps between levels:

≠0 - on, (function unknown).

=0 - off.

Col. 29-32 - Multipath sensitization:

≠0 - on, (function unknown).

=0 - off.

Col. 33-36 - Derace option:

=0 - on, an attempt is made to insert test patterns necessary to ensure that the translation between one pattern and the next does not result in a race condition.

≠0 - off.

Col. 37-40 - Don't care state assignment:

=0 - all don't care input states will be set to 0.

=1 - all don't care input states will be set to 1.

=2 - all don't care input replacement states are to be read from the next data card in 80I1 format.

=blank - don't care states will be alternated.

Col. 41-44 - Supermain override:

=0 - on, (function unknown).

$\neq 0$ - off.

Col. 45-48 - Print option reset value:

=value - same interpretation as for the primary print option and is used as the reset value for the primary print option when the print option reset flag is set.

Col. 49-52 - (Function unknown).

Col. 53-56 - (Function unknown).

Col. 57-60 - Derace input skew value:

=value - gate delay time multiple used with the derace option. The value is applied as a time skew on inputs. Pattern sets can be deraced where a time to the skew value of gate delay might re-result in a race condition.

Col. 61-64 - Print option reset flag:

≤ 0 - number of detections after which the primary print option is switched to the print option reset value and the secondary print option is reset to 2.

=0 - off.

Col. 65-68 - Maximum time allowed for completion of a set of stimuli:

=value - maximum time in seconds.

=0 - reset to 120.

B.1.2 Specification of Untestable Node States (ILL Card)

This optional input allows the user to input node states, or sets of node states, which the program will not be allowed to force in generating stimuli. The user specifies these illegal node states in terms of the user node numbers (internal component numbers may not be used), with the sign indicating the illegal state; i.e., positive = 1, negative = 0. The following is the card format to be used:

Col. 1-3	-	ILL (required).
-13 five-column fields (first field = 6-10, last field = 66-70).	-	illegal node state specifications, right-justified in the field. One or more blank fields delimits an illegal set.

The following example indicates that node 100 is illegal in the 0 state and node 150 at 1 and node 151 at 0 are an illegal set of node states.

ILLbbb-100bbbbbbb150b-151

All ILL cards must follow the option card and any don't care skeleton cards in the input deck.

B.1.3 Specification of Response Patterns (INIT Cards)

This option input allows the user to specify individual nodes or sets of nodes as response patterns for generating stimuli. This is done by entering a negative number on the option card (columns 17-20), and using special response initial cards. Each response pattern is specified as a list of user node numbers (internal component node numbers may not be used), where the sign of each indicates the node's state; i.e., positive = 1, negative = 0. The following is the required card format:

Col. 1-4	-	INIT (required).
-13 five-column fields (first field = 6-10, last field = 66-70).	-	response pattern specifications, right-justified in the fields, no internal blank fields.

Each response pattern must begin on an INIT card.

Continuation cards may be used when necessary. They are of the same form as the INIT cards.except that columns 1-4 are left blank.

The following is an example of a response pattern specification:

INITbbb-13bbb14

indicating that node 13 at 0 and node 14 at 1 are to be a response pattern for each stimuli are to be generated (i.e., they are set up as output initials).

All response pattern (INIT) cards must follow any ILL cards, at the end of the stimulus generation data deck.

APPENDIX C

EDIT USER'S MANUAL

Report No. 72-0025
Contract No. NAS8-25621

EDIT USER'S MANUAL

August 22, 1972

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Marshall Space Flight Center
Huntsville, Alabama

M&S COMPUTING, INC.

PREFACE

This document serves as the User's Manual for the EDIT utility program operating on the Xerox Data Systems' Sigma 5 computer. The EDIT program was written under Marshall Space Flight Center Contract No. NAS8-25621 to facilitate the updating of large FORTRAN programs stored in source card format in a magnetic tape library. It will serve as a useful utility, however, for editing any card image format records.

This manual presumes the EDIT user to have a basic knowledge of the Sigma 5 system control cards required for FORTRAN program execution and I/O device assignment. It contains a discussion of EDIT operations, definitions of all control commands, examples of run decks, and explanations of printouts.

Prepared by:

Robert L. Kuehlth[illegible]

Approved by:

J. W. Meadlock

TABLE OF CONTENTS

1. INTRODUCTION

EDIT is a utility program written in XDS extended FORTRAN to update files of card image records. It is capable of inserting and deleting records within files, manipulating multifile tapes, reading and writing blocked files, and labeling and sequencing card image records. These operations are controlled by a file of control commands which have an unusually simple syntax, minimizing the effort required to prepare control decks for complicated file editing. The control commands of this file are processed sequentially until the end-of-file is encountered and execution is terminated.

All I/O operations are fully buffered, permitting simultaneous instruction execution and I/O on several devices. Physical records of any length may be read by the input unit, logical unit 1, but all data is formatted into 80 character card image records before update processing. Unit 1 physical records of less than 80 characters are padded with blanks while physical records of up to 1600 will be broken into consecutive 80 character records. Only the first 1600 characters of larger physical records will be processed. Updating of the 80 character records proceeds as directed by the command file, and the modified data is written on all selected output files. Control commands allow the user to enable or disable output on the line printer, logical unit 2, and logical unit 3. Data output on unit 2 is blocked 1600 characters per physical record, while data on unit 3 remains in card image form.

2. EDIT CONTROL FILE

When EDIT execution begins one file of data is copied from logical unit 105 (assigned to the card reader by default) to temporary storage for interpretation as the control command file. The records of this file will be interpreted as control commands if their first character is a minus sign. All other records are interpreted as update records to be inserted into the output file. The control commands are of two basic types: locators, which specify positions within a file; and operations, which initiate file manipulations and other utility tasks. Locators are of the form:

-n

or

-n, m

where n and m are integers in XDS free format which designate positions of records in the input file being processed. The first form instructs the program to copy n records from input to output files. The second form instructs EDIT to copy input to output <u>until</u> the n^{th} record of the input file is encountered. Records n through m of the input file are then skipped (effectively deleted from the output file), and the input file is positioned to resume copying on the $m+1^{th}$ record if another copy command is executed. It should be emphasized that the locator numbers n and m specify 80 character record positions (sequence counts) within the input file being processed. Since update processing is irreversible, locators of several positions within a given input file must always be ordered on increasing n. Similarly, m must be greater than or equal to n and n must not be greater than the number of records contained in the input file.

Operation control commands are executed as they are encountered in the control file. Update records (those which do not begin with a minus sign) are copied directly to the output files when they are encountered in the control file. Thus the operations and updates which follow a locator are processed in sequence until another locator is encountered in the control stream. Operations and updates may be processed at the beginning of a file by placing them ahead of any locator for that file.

Table 2-1 defines the operation control commands which are available to the EDIT user. The asterisked commands are default options which are in effect at the start of execution.

OPERATION CONTROL COMMANDS

-CPF n	**Copy through n end-of-files on unit 1. (If n is omitted, it is assumed to be 1.)**
-CPA	**Copy through a double end-of-file (2 successive end-of-files) on unit 1.**
-SKP n	**Position unit 1 on the first record following n end-of-files. (If n is omitted, it is assumed to be 1.)**
-EOF	**Write an end-of-file on all output files.**
-REW n	**Rewind logical unit n, where n=1, 2, or 3.**
-DEF n	**Define all update card images between this and the next control command to be elements of "include" block n, where: $0 \leq n \leq 5$ and the omission of n implies n=0.**
-INC n	**Copy all the card images of previously defined include block n to the output files.**
-ON 2	**Write subsequent output file data to logical unit 2. Data output on unit 2 is normally blocked 1600 characters per physical record. The last block of a file, however, will contain only those characters which complete the card image contents of that file.**
-ON 3	*** Write subsequent output file data to logical unit 3 in 80 character card images.**
-OFF 2	*** Inhibit the writing of subsequent output file data to logical unit 2.**
-OFF 3	**Inhibit the writing of subsequent output file data to logical unit 3.**
-PR	Write subsequent output file data to the line printer.
-NPR	* Inhibit the writing of subsequent output file data to the line printer.
-LB cde	Replace characters 73 through 75 of each output file record with characters c, d, and e.

Table 2-1

OPERATION CONTROL COMMANDS

(continued)

-NLB	*** Disable the replacement of characters 73 through 75 of the output file records.**
-SEQ	**Replace characters 77 through 80 of each output file record with the least significant four digits of the record's position count in the output file.**
-NSQ	*** Disable the replacement of characters 77 through 80 of the output file records.**

Table 2-1
(continued)

The numbers requested by many of the operation commands are integers extracted from characters 5 through 80 of the command in XDS free format. There is no limit to the size of the files processed by EDIT, but the number of card images defined for any include block may not exceed 100.

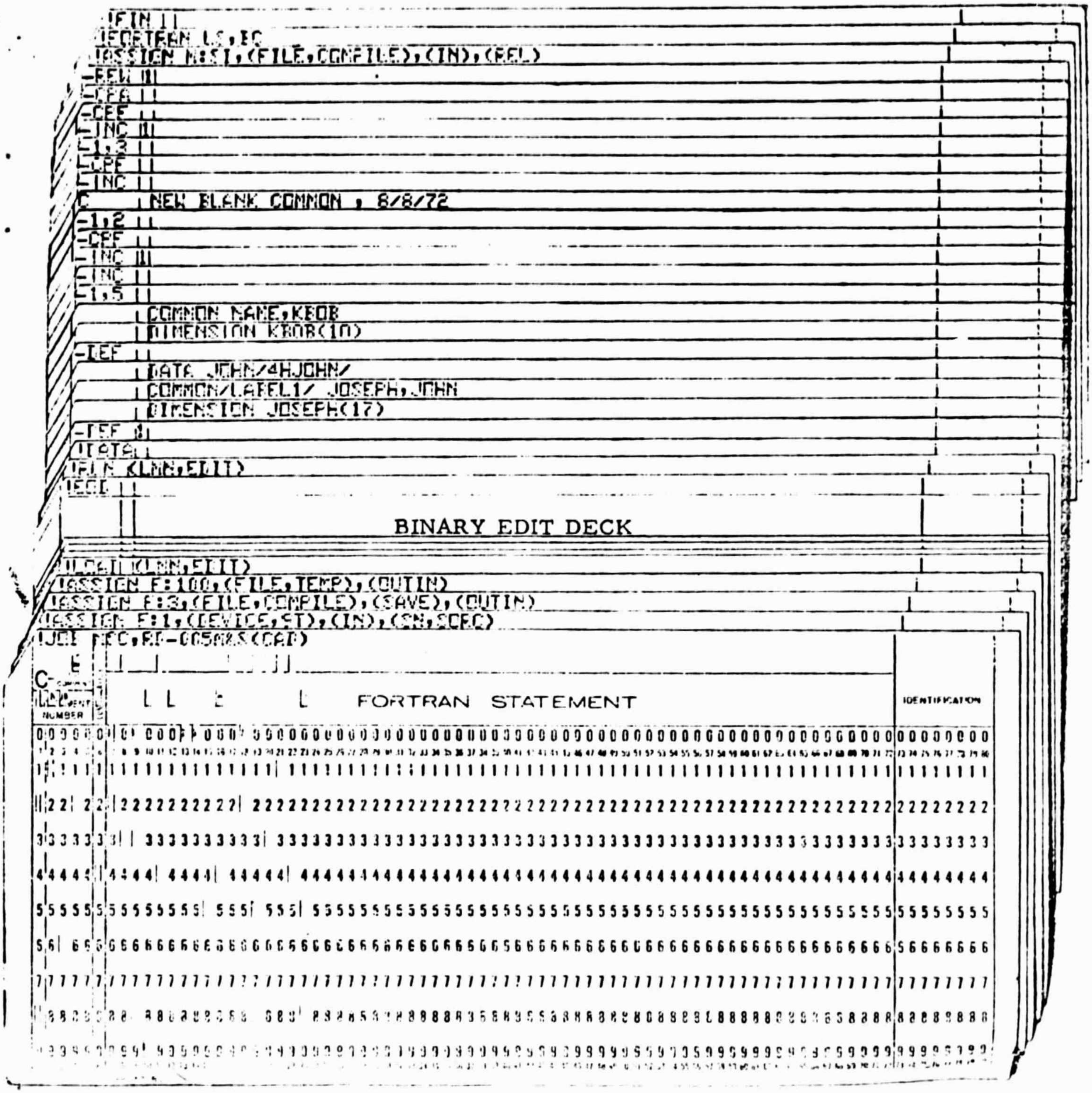
3. SAMPLE EDIT RUN DECKS

The user must assign a device or file to logical unit 100 and to logical units 1, 2, and 3 when these are to be used by his EDIT run. Unit 100 is required for scratch storage (OUTIN), and is normally assigned as a temporary disk file. Unit 1 will be used for input (IN), while units 2 and 3 are to be used in the scratch mode (OUTIN) since they are backspaced when write errors occur.

EDIT has been designed to facilitate frequent updates of a master source tape and to support generation of a new source tape when a number of updates have accumulated. It is suggested that master source tapes be blocked for speed of access and sequenced to simplify update procedure. Compiler or assembler listings may then be used to obtain sequence numbers for updates. Figure 3-1 illustrates a deck which may be used to load EDIT from a binary BO deck, create an updated compiler input file from a master source tape, and compile that file. This example shows how the include blocks may be intermixed with update statements for insertion and how these may be used to change the common definitions in sub-programs stored on separate files.

The example of Figure 3-2 illustrates the use of EDIT to simultaneously generate a compiler input file and an updated master source tape. Note that the comment card will be copied into the output before any of the original records since it precedes all locator commands for the file. The second edit run of this example will read the new master tape and thereby verify that it has no errors.

New source tapes may be created from cards by assigning logical unit 1 to the card reader as illustrated in Figure 3-3. Note that the control command file precedes the input data deck and is separated from it by an end-of-file card. The user should be aware of one unique restriction imposed when card input is used as in this example: the entire input file must be read to an end-of-file before EDIT program termination. Thus a -CPF or a -CPA must be the last locator command.



NEW BLANK COMMON , 8/8/72
COMMON NAME,KBOB
DIMENSION KBOB(10)
DATA JOHN/4HJOHN/
COMMON/LABEL1/ JOSEPH,JOHN
DIMENSION JOSEPH(17)
BINARY EDIT DECK
!ASSIGN F:100,(FILE,TEMP),(OUTIN)
!ASSIGN F:3,(FILE,COMPILE),(SAVE),(OUTIN)
FORTRAN STATEMENT
IDENTIFICATION

Figure 3-1

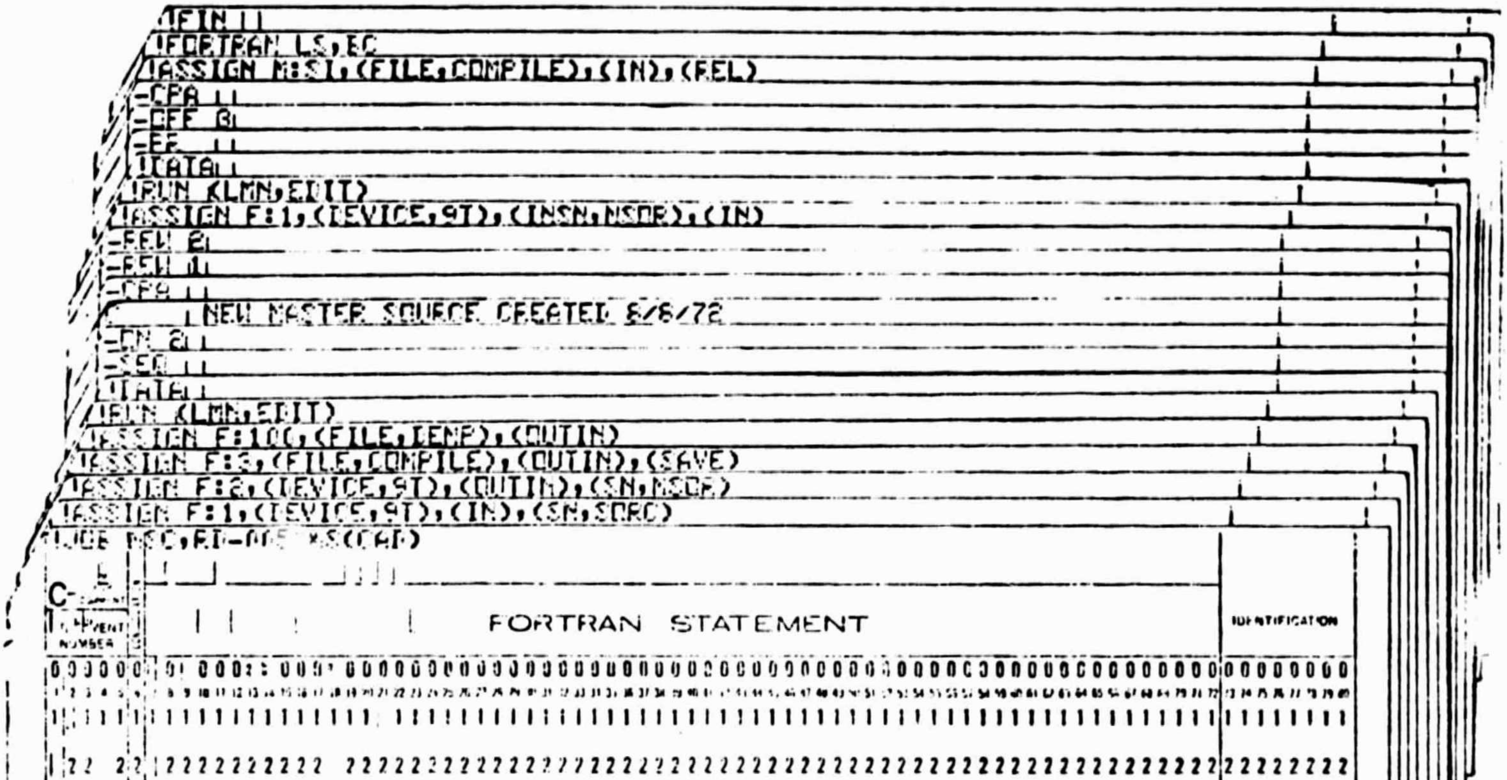

!FIN
!FORTRAN LS,EC
!ASSIGN M:SI,(FILE,COMPILE),(IN),(REL)
-CPA
-EE
!DATA
!RUN (LMN,EDIT)
!ASSIGN F:1,(DEVICE,9T),(INSN,MSDR),(IN)
-REW 2
-CPA
NEW MASTER SOURCE CREATED 8/8/72
-CN 2
-SEQ
!DATA
!RUN (LMN,EDIT)
!ASSIGN F:100,(FILE,TEMP),(OUTIN)
!ASSIGN F:3,(FILE,COMPILE),(OUTIN),(SAVE)
!ASSIGN F:2,(DEVICE,9T),(OUTIN),(SN,MSDR)
!ASSIGN F:1,(DEVICE,9T),(IN),(SN,SDRC)
FORTRAN STATEMENT
IDENTIFICATION

Figure 3-2

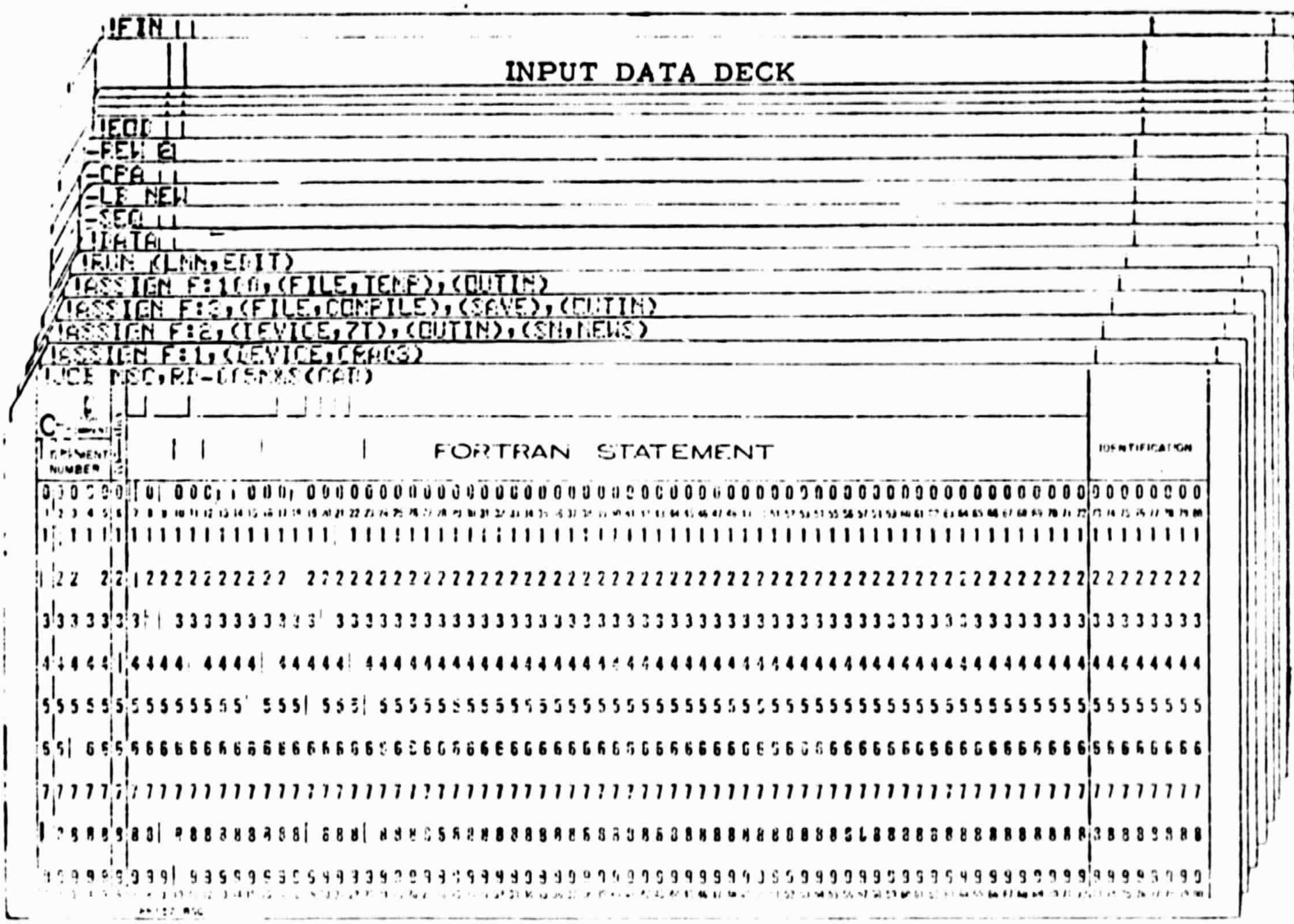

!FIN
INPUT DATA DECK
!EOD
!RUN (LMN,EDIT)
!ASSIGN F:100,(FILE,TEMP),(OUTIN)
FORTRAN STATEMENT

Figure 3-3

4. EDIT PRINTOUT

Normal EDIT printout consists of a listing of the control command file, the optional printout of the output file with sequence numbers and flags identifying the updates, and a summary of I/O errors from which the program has recovered. The following lines will be printed if an illegal operation command is encountered or a locator command is not placed in proper sequence for the file being updated:

ILLEGAL COMMAND:
xxxxxxxxxxxxxxxxxxxxxxxxx
STATEMENT IGNORED

where the x's are filled by the characters of the offending statement. Since an out of sequence locator must be ignored, the updates and operations which follow are usually located incorrectly in the output.

If an error occurs while writing to an output unit, that unit is backspaced and another write is attempted. When five successive attempts fail, an empty or vacuous record is written to attempt to skip damaged area of output tapes. Such records will be ignored if the output tape is to be read only by this EDIT program, but other applications may not accept such records. The following message informs the user each time a vacuous record has been written:

WRITE ERROR ON UNIT n
VACUOUS RECORD INSERTED TO SKIP BAD SPOT

where n is the logical output unit number.

If an error occurs while writing a vacuous record, the system will abort the job. Fatal I/O errors result when a read is unsuccessful after 50 successive attempts or when 10 successive vacuous records do not circumvent write errors. The following message is printed when execution is terminated for either of these conditions:

ERROR TERMINATION - EXCESSIVE I/O ERRORS ON UNIT n

where n is the logical unit number causing the termination.

REFERENCES

1. Banning PRF Programmer's Manual, Report No. 70-0036, M&S Computing, Inc., December 30, 1970.

2. Banning Placement-Routing-Folding User's Manual, Report No. 70-0021, M&S Computing, Inc., July 31, 1970.

3. Printed Wiring Board System User's Manual, Report No. 73-0014, M&S Computing, Inc., Revision 1, June 4, 1973.

4. Printed Wiring Board System Programmer's Manual, Report No. 73-0013, M&S Computing, Inc., April 25, 1973.

5. LOGSIM User's Manual, Report No. 72-0001, M&S Computing, Inc., January 13, 1972.

6. LOGSIM Programmer's Manual, Report No. 72-0002, M&S Computing, Inc., January 18, 1972.

7. FETLOG User's Manual, to be published.

8. MAP User's Manual, to be published.

9. Artwork Interactive Design System - AIDS - Program Description, Report No. 73-0002, M&S Computing, Inc., January 12, 1973.

10. Banning Artwork Program User's Manual, Report No. 70-0012, M&S Computing, Inc., April 24, 1970.

11. Programmer's Manual for the Banning Artwork Program, Report No. 70-0013, M&S Computing, Inc., June 5, 1970.

12. NTRAN User's Manual, Report No. 75-0028, M&S Computing, Inc., July 30, 1975.